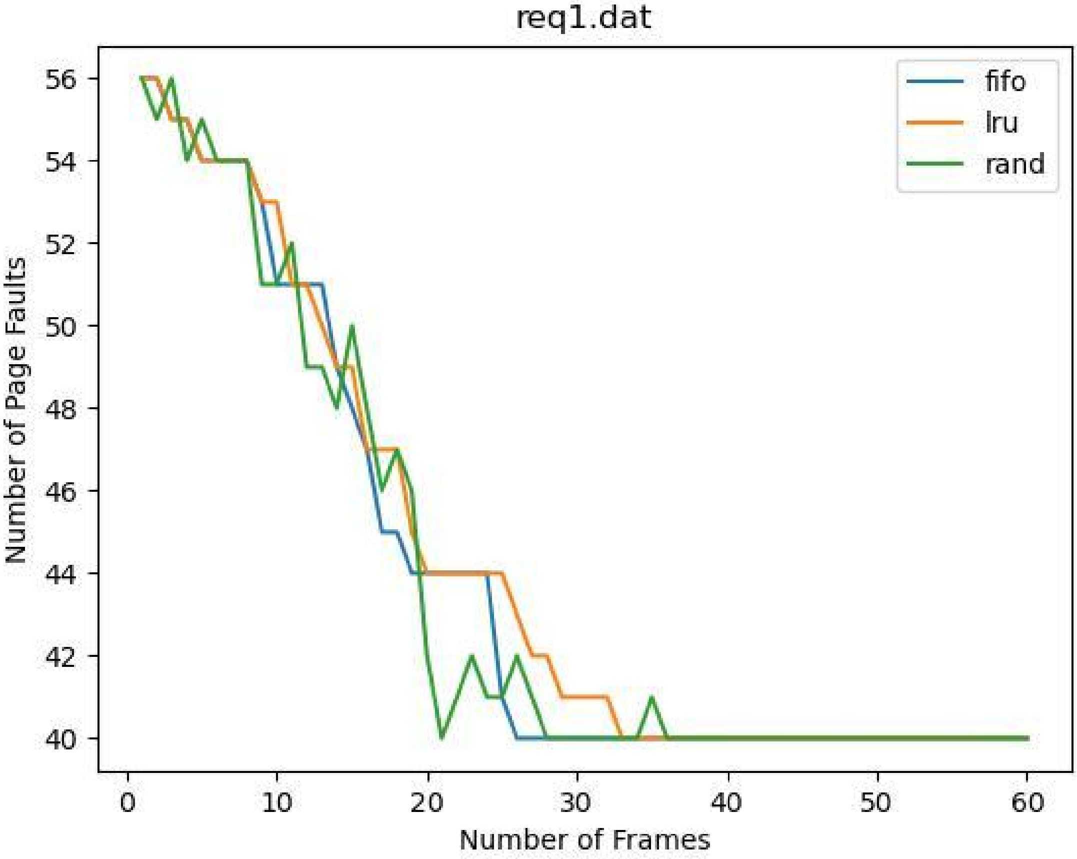
Operating Systems Laboratory

Lab 8

Om Patil (200010036)

req1.dat

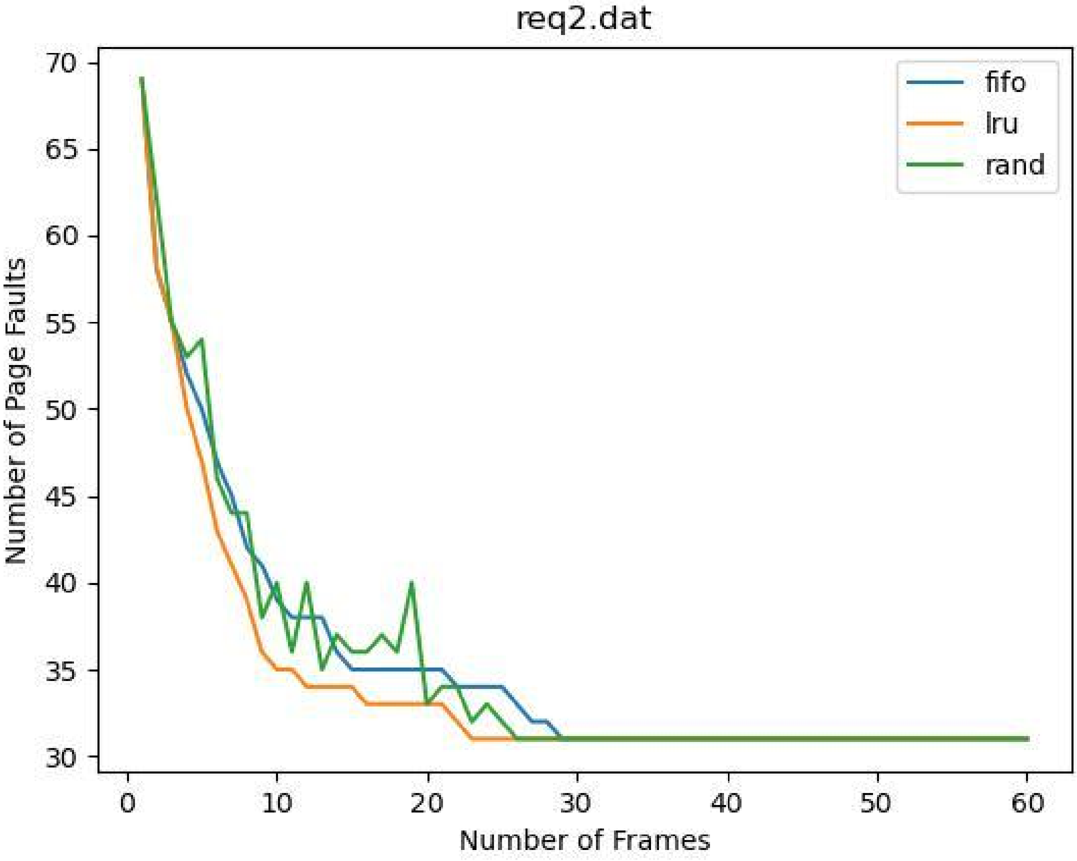


The configuration used to simulate the above has 60 addressable pages

Observations:

As we can see increasing, the number of frames available reduces the number of page faults in general, regardless of the replacement policy. After a point, increasing the number of frames doesn’t reduce the number of page faults, as there is enough space in memory such that nearly no swapping occurs. The final value for the number of page faults it settles at is 40. These page faults are unavoidable faults that occur at the start when the memory is empty. We can also observe that the random policy is quite good, but its performance is highly uncertain.

req2.dat



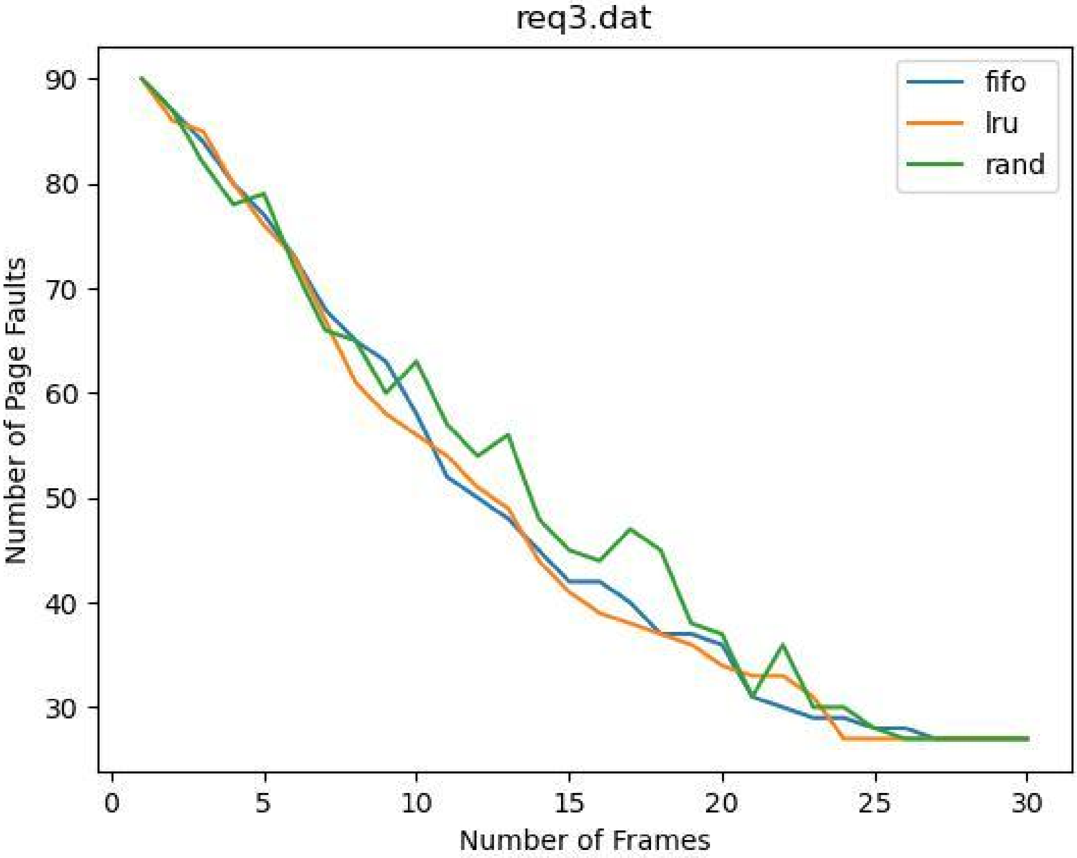
The configuration used to simulate the above has 60 addressable pages. In this request file, requests are generated randomly. However, certain frames are generated much more often than others. This better simulates a real request sequence.

Observations:

As we can see increasing, the number of frames available reduces the number of page faults in general, regardless of the replacement policy. After a point, increasing the number of frames doesn’t reduce the number of page faults, as there is enough space in memory such that nearly no swapping occurs. The final value for the number of page faults settles around 31. These page faults are unavoidable faults that occur at the start when the memory is empty. We can also observe that the random policy is quite good, but its performance is highly uncertain.

Due to the nature of the requests, LRU performs quite well in this request file. This is because LRU avoids swapping the frames, which are generated quite often. This leads to fewer page faults.

req3.dat



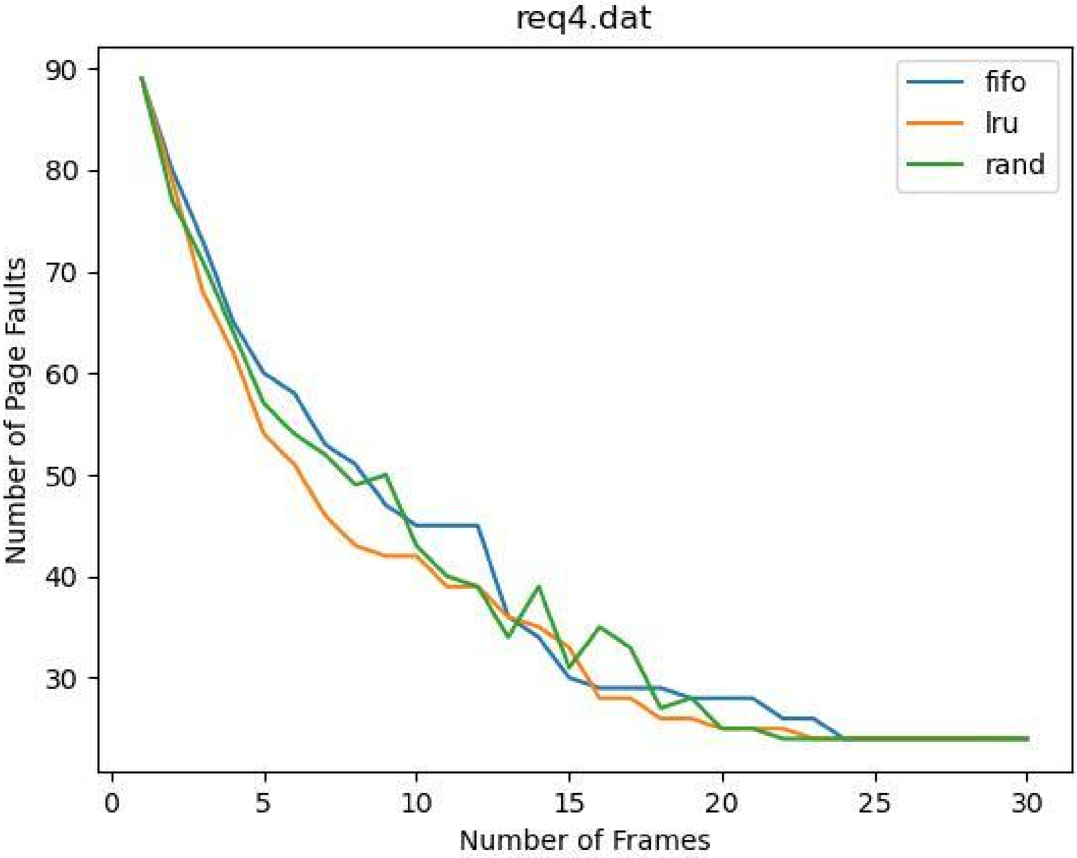
The configuration used to simulate the above has 30 addressable pages. In this request file, requests are generated completely randomly.

Observations:

As we can see increasing, the number of frames available reduces the number of page faults in general, regardless of the replacement policy. After a point, increasing the number of frames doesn’t reduce the number of page faults, as there is enough space in memory such that nearly no swapping occurs. The final value for the number of page faults settles around 29. These page faults are unavoidable faults that occur at the start when the memory is empty. We can also observe that the random policy is quite good, but its performance is highly uncertain.

Due to the random nature of the requests, all the replacement policies perform similarly. Also, due to the smaller number of addressable pages, the number of page faults settle down closer to the maximum number of frames.

req4.dat



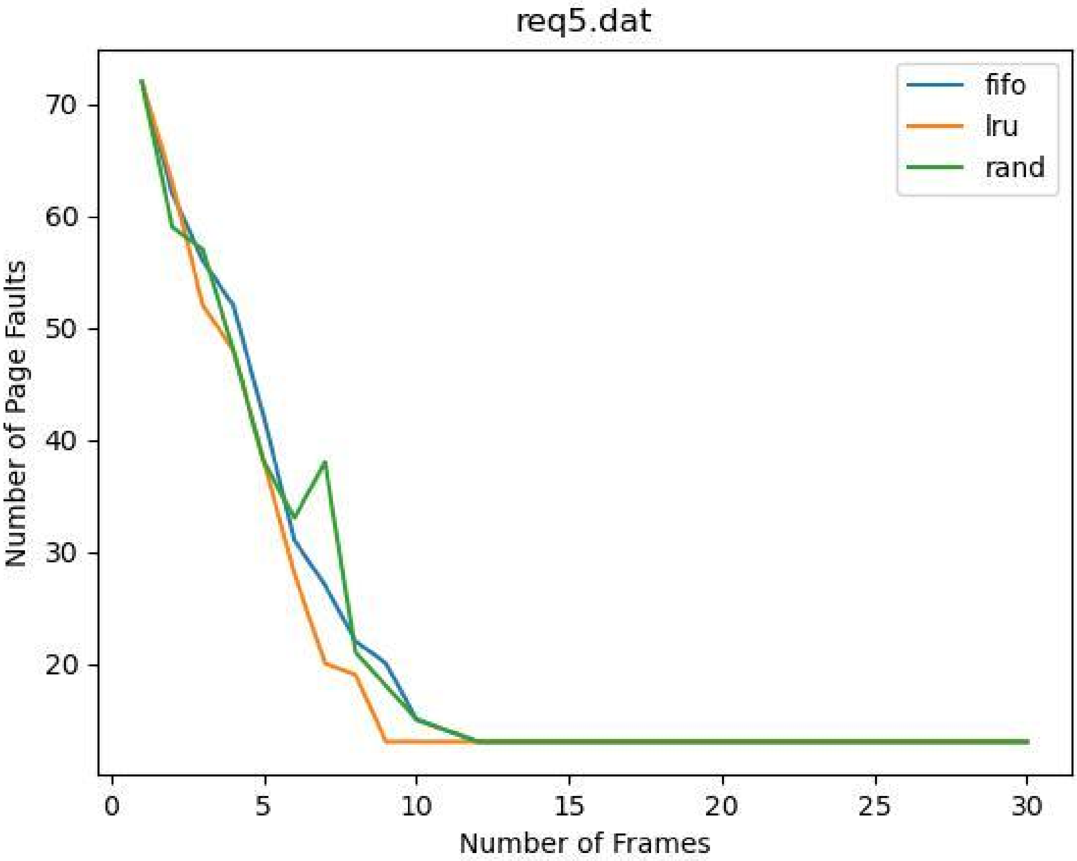
The configuration used to simulate the above has 30 addressable pages. In this request file, requests are generated randomly. However, certain frames are generated much more often than others. This better simulates a real request sequence.

Observations:

As we can see increasing, the number of frames available reduces the number of page faults in general, regardless of the replacement policy. After a point, increasing the number of frames doesn't reduce the number of page faults, as there is enough space in memory such that nearly no swapping occurs. The final value for the number of page faults settles around 31. These page faults are unavoidable faults that occur at the start when the memory is empty. We can also observe that the random policy is quite good, but its performance is highly uncertain.

Due to the nature of the requests, LRU performs quite well in this request file. This is because LRU avoids swapping the frames, which are generated quite often. This leads to fewer page faults. Also, due to the smaller number of addressable pages, the number of page faults settle down closer to the maximum number of frames.

req5.dat



The configuration used to simulate the above has 30 addressable pages. In this request file, requests are generated randomly. However, certain frames are generated more often than others by sampling from a normal distribution. This better simulates a real request sequence.

Observations:

As we can see increasing, the number of frames available reduces the number of page faults in general, regardless of the replacement policy. After a point, increasing the number of frames doesn’t reduce the number of page faults, as there is enough space in memory such that nearly no swapping occurs. The final value for the number of page faults settles around 31. These page faults are unavoidable faults that occur at the start when the memory is empty. We can also observe that the random policy is quite good, but its performance is highly uncertain.

Due to the nature of the requests, LRU performs quite well in this request file. This is because LRU avoids swapping the frames, which are generated quite often. This leads to fewer page faults. Sampling from a normal distribution causes only a certain subset of the frames to be used often. This leads to the number of faults settling down quite early. This is because fewer frames are required, as very few frames must be swapped.